



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/917,397	07/27/2001	Richard Glenn Goodwin	70662	1536

22242 7590 09/08/2005

FITCH EVEN TABIN AND FLANNERY  
120 SOUTH LA SALLE STREET  
SUITE 1600  
CHICAGO, IL 60603-3406

EXAMINER

FOWLKES, ANDRE R

ART UNIT	PAPER NUMBER
----------	--------------

2192

DATE MAILED: 09/08/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

87

## Office Action Summary

Application No.

09/917,397

Applicant(s)

GOODWIN ET AL.

Examiner

Andre R. Fowlkes

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 18 April 2005.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-64 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-64 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

## DETAILED ACTION

1. This action is in response to the amendment filed 4/18/05.

### *Double Patenting*

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 21-27 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-7 of U.S. Patent No.

6,199,195. Although the conflicting claims are not identical, they are not patentably distinct from each other because they are obvious variations of each other. For

example, claim 21 of the instant application recites "**A method for generating source code objects comprising: generating a logical model using a modeling tool; translating the logical model into a corresponding unified model; generating a system definition comprising a template, the template defining at least one service within a framework; and generating at least one source code object as a**

Art Unit: 2192

function of the unified model, and the template". Claim 1 of the conflicting patent recites "A method for generating source code objects comprising:

**generating a plurality of logical models using a plurality of modeling tools;**

**translating each of the plurality of logical models into corresponding ones of a plurality of unified models; generating a system definition comprising a plurality of templates, each defining at least one service within a framework; and generating at least one source code object as a function of at least one of said plurality of unified models, and at least one of said plurality of templates".**

Therefore, it would have been an obvious variation, to a person of ordinary skill in the art, at the time the invention was made, to generate a plurality of models using a plurality of modeling tools and incorporate a plurality of templates in the system definition. The modification would have been obvious because one of ordinary skill in the art would have wanted the flexibility of generating a plurality of models using a plurality of modeling tools and incorporate a plurality of templates in the system definition to develop an optimum product. Additionally, one of ordinary skill in the art would have wanted to generate a single model using a single modeling tool, to conserve memory and time.

### ***Claim Rejections - 35 USC § 102***

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

Art Unit: 2192

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-64 are rejected under 35 U.S.C. 102(e) as being anticipated by Iyengar et al., (Iyengar), U.S. Patent No. 6,018,627.

As per claim 1, Iyengar discloses **a method for generating source code**, (col. 2:62-63, "the system generates ... code based on the (translated) UML model"),  
**comprising:**

- **generating a translation file containing translation logic** (col. 1:33-35, "The process may start with the building of business models and progress to representing the business models as object models (i.e. translate the business models using translation logic)"),

- **inputting the translation file into a code generator and generating translation source code as a function of the translation file** (col. 1:33-35, "The process may start with the building of business (i.e. logic) models and progress to representing the business models as object models (i.e. UML)", and col. 2:62-63, "the system generates ... code based on the (translated) UML model").

As per claim 2, the rejection of claim 1 is incorporated and further, Iyengar discloses **generating a plurality of translation files; and inputting the plurality of**

Art Unit: 2192

**translation files into the code generator** (col. 1:33-35, "The process may start with the building of business (i.e. logic) models and progress to representing the business models as object models (i.e. UML)", and col. 2:62-63, "the system generates ... code based on the (translated) UML model").

As per claim 3, the rejection of claim 1 is incorporated and further, lyengar discloses **generating a logical model using a modeling tool; translating the logical model into a unified model; and inputting the unified model into the code generator** (col. 1:33-35, "The process may start with the building of business (i.e. logical) models and progress to representing the business models as object models (i.e. UML)", and col. 2:62-63, "the system generates ... code based on the UML model").

As per claim 4, the rejection of claim 3 is incorporated and further, lyengar discloses **generating at least one source code object as a function of the unified model** (col. 2:62-63, "the system generates ... code based on the UML model").

As per claim 5, the rejection of claim 3 is incorporated and further, lyengar discloses **generating of at least one source code object comprises generating at least one interface definition language element** (col. 2:62-63, "the system generates ... code based on the UML model (including IDL elements)").

As per claim 6, the rejection of claim 3 is incorporated and further, Iyengar discloses **storing the unified model in a schema repository; and storing the translation file in the schema repository** (col. 2:36-37, "once the output data is transformed into a UML model, it is saved in a repository", and col. 9:2-6, "Under another method, data from the modeling tool is exported into a database, and the database is saved into the repository. Data from the database is then imported into a UML model by means of the Transformation Algorithms").

As per claim 7, Iyengar also discloses such claimed limitations as addressed in claim 6 above.

As per claim 8, the rejection of claim 1 is incorporated and further, Iyengar discloses that **the translation file contains translation logic to translate data from a database into a standard format** (col. 9:2-6, "Under another method, data from the modeling tool is exported into a database, and the database is saved into the repository. Data from the database is then imported into a UML model by means of the Transformation Algorithms").

As per claim 9, Iyengar discloses **a method of generating source code objects**, (col. 2:62-63, "the system generates ... code based on the (translated) UML model"), **comprising:**

- **providing a code generator with a unified model represented in a unified modeling language** (col. 1:33-35, "The process may start with the building of business models and progress to representing the business models as object models (i.e. UML)", and col. 2:62-63, "the system generates ... code based on the (translated) UML model").

- **providing the code generator with a system definition** (col. 1:33-35, "The process may start with the building of business models (i.e. a system definition) and progress to representing the business models as object models (i.e. UML)", and col. 2:62-63, "the system generates ... code based on the (translated) UML model").

- **providing the code generator with a translation file in the unified modeling language** (col. 1:33-35, "The process may start with the building of business models and progress to representing the business models as object models (i.e. translate the business models using translation logic)"),

- **generating source code objects as a function of the unified model and the system definition** (col. 1:33-35, "The process may start with the building of business models (i.e. a system definition) and progress to representing the business models as object models (i.e. UML)", and col. 2:62-63, "the system generates ... code based on the (translated) UML model"),

- **generating source code objects as a function of the unified model and the translation file** (col. 1:33-35, "The process may start with the building of business models and progress to representing the business models as object models (i.e.



translate the business models using translation logic) and col. 2:62-63, "the system generates ... code based on the (translated) UML model").

As per claims 10-12, this is another method version of the claimed method discussed above, in claims 1-4, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see Iyengar's system for application building in an object oriented environment with OMG compliant UML data stored in a repository (col. 2:27-3:8).

As per claims 13-20, this is a system version of the claimed method discussed below, in claims 27 and 29, wherein all claimed limitations have also been addressed and/or cited as set forth below. For example, see Iyengar's system for application building in an object oriented environment with OMG compliant UML data stored in a repository (col. 2:27-3:8).

As per claim 21, Iyengar discloses a **method for generating source code objects**, (col. 2:62-63, "the system generates ... code based on the UML model"), **comprising:**

- **generating a logical model using a modeling tool** (col. 1:33-35, "The process may start with the building of business (i.e. logical) models and progress to representing the business models as object models (i.e. unified model)"),

- **translating the logical model into a corresponding unified model** (col. 1:33-35, "The process may start with the building of business (i.e. logical) models and progress to representing the business models as object models (i.e. unified model)"),

- **generating a system definition comprising a template, the template defining at least one service within a framework** (col. 1:33-35, "The process may start with the building of business models (i.e. template) and progress to representing the business models as object models (i.e. unified model)"),

- **generating at least one source code object as a function of the unified model, and the template** (col. 1:33-35, "The process may start with the building of business (i.e. template) models and progress to representing the business models as object models (i.e. UML)", and col. 2:62-63, "the system generates ... code based on the UML model").

As per claim 22, the rejection of claim 21 is incorporated and further, Iyengar discloses that **translating of the logical model comprises generating at least one Unified Modeling Language (UML) element** (col. 1:33-35, "The process may start with the building of business (i.e. template) models and progress to representing the business models as object models (i.e. UML)").

As per claim 23, the rejection of claim 21 is incorporated and further, Iyengar discloses that **the generating of the system definition comprises generating in the template at least one JavaScript element** (col. 10:2, "Java").

As per claim 24, the rejection of claim 21 is incorporated and further, Iyengar discloses that **generating of at least one source code object comprises generating at least one interface definition language element** (col. 9:18, "UML (includes IDL elements)").

As per claim 25, the rejection of claim 21 is incorporated and further, Iyengar discloses that **the adaptor defining a translation from the modeling tool** (col. 9:2-6, "Under another method, data from the modeling tool is exported into a database (using an adapter), and the database is saved into the repository. Data from the database is then imported into a UML model by means of the Transformation Algorithms").

As per claim 26, the rejection of claim 21 is incorporated and further, Iyengar discloses that **storing the unified model in a schema repository; wherein said generating of the source code objects comprises retrieving the unified model** (col. 2:36-37, "once the output data is transformed into a UML model, it is saved in a repository").

As per claim 27, the rejection of claim 21 is incorporated and further, Iyengar discloses **retrieving data from a database by employing the source code objects and the unified model to define a relationship between an object oriented database query and the data** (col. 2:36-37, "once the output data is transformed into a

Art Unit: 2192

UML model, it is saved in a repository”, and col. 9:2-6, “Under another method, data from the modeling tool is exported into a database, and the database is saved into the repository. Data from the database is then imported into a UML model by means of the Transformation Algorithms”).

As per claim 28, the rejection of claim 21 is incorporated and further, Iyengar discloses **generating a translation file, the translation file containing translation logic; and generating at least one source code object as a function of the translation file** (col. 9:7-12, “The next step of the development flow may be the creation of object models. Object models may be constructed or modified during the domain modeling 27 process from transformed legacy items or from enterprise models. The system, moreover, allows reverse engineering of object models into enterprise models”).

As per claim 29, the rejection of claim 21 is incorporated and further, Iyengar discloses that **storing the translation file in a schema repository** (col. 9:2-6, “Under another method, data from the modeling tool is exported into a database, and the database is saved into the repository. Data from the database is then imported into a UML model by means of the Transformation Algorithms”).

As per claims 30-35, this is another method version of the claimed method discussed above, in claims 21-24, 27 and 29, wherein all claimed limitations have also

been addressed and/or cited as set forth above. For example, see Avengers system for application building in an object oriented environment with OMG compliant UML data stored in a repository (col. 2:27-3:8).

As per claims 36-43 and 46, this is a system version of the claimed method discussed above, in claims 21-27, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see Iyengars system for application building in an object oriented environment with OMG compliant UML data stored in a repository (col. 2:27-3:8).

As per claim 44, the rejection of claim 36 is incorporated and further, Iyengar discloses **Java RMI** (col. 10:2, "Java (RMI)").

As per claim 45, the rejection of claim 36 is incorporated and further, Iyengar discloses **OMG framework** (13:22, "OMG (framework)").

As per claims 47-51, this is another method version of the claimed method discussed above, in claims 21-24, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see Iyengars system for application building in an object oriented environment with OMG compliant UML data stored in a repository (col. 2:27-3:8).

Art Unit: 2192

As per claims 53-58, this is another method version of the claimed method discussed above, in claims 21-27, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see Iyengar's system for application building in an object oriented environment with OMG compliant UML data stored in a repository (col. 2:27-3:8).

As per claims 59-64, this is a system version of the claimed method discussed above, in claims 21-27, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see Iyengar's system for application building in an object oriented environment with OMG compliant UML data stored in a repository (col. 2:27-3:8).

### ***Response to Arguments***

6. Applicants' arguments have been considered but they are not persuasive.

*In the remarks, the applicant has argued substantially that:*

- 1) Iyengar does not disclose generating a translation file containing translation logic, at p. 16:10-17:2 & 19:1-12.

*Examiner's response:*

- 1) The examiner disagrees with applicant's characterization of the applied art. Iyengar does disclose generating a translation file containing translation logic at col.

1:33-35, "The process may start with the building of business models and progress to representing (i.e. translate) the business models as object models", as discussed in the above art rejection.

*In the remarks, the applicant has argued substantially that:*

- 2) lyengar does not disclose generating source code, at p. 17:3-5.

*Examiner's response:*

- 2) The examiner disagrees with applicant's characterization of the applied art. lyengar does disclose generating source code at col. 9:50, "visual basic skeleton code", as discussed in the above art rejection.

*In the remarks, the applicant has argued substantially that:*

- 3) lyengar does not disclose generating translation source code as a function of a translation file, at p. 17:15-18 and p. 19:14-25.

*Examiner's response:*

- 3) The examiner disagrees with applicant's characterization of the applied art. lyengar does disclose generating translation source code at col. 1:33-35, "The process may start with the building of business (i.e. logic) models and progress to representing the business models as object models (i.e. UML translation file)", and col. 2:62-63, "the system generates ... code based on the (translated) UML model" as discussed in the above art rejection.

Art Unit: 2192

*In the remarks, the applicant has argued substantially that:*

4) Iyengar does not teach generating source code objects as a function of the unified model and the system definition (comprising a template), at p. 18:12-20 & 20:14-20.

*Examiner's response:*

4) The examiner disagrees with applicant's characterization of the applied art. Iyengar does disclose generating source code objects as a function of the unified model and the system definition (comprising a template) at (col. 1:33-35, "The process may start with the building of business models (i.e. templates) and progress to representing the business models as object models) and col. 2:62-63, "the system generates ... code based on the UML model (composed from the template)"), as discussed in the above art rejection.

### ***Conclusion***

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andre R. Fowlkes whose telephone number is (571) 272-3697. The examiner can normally be reached on Monday - Friday, 8:00am-4:30pm.



Art Unit: 2192

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571)272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

ARF



TUAN DAM  
SUPERVISORY PATENT EXAMINER